Training the Stochastic Kinetic Model of Neuron for Calculation of an Object's Position in Space

Aleksandra Świetlicka¹ · Krzysztof Kolanowski¹ · Rafał Kapela¹

Received: 19 February 2019 / Accepted: 15 July 2019 / Published online: 31 July 2019
© The Author(s) 2019

Abstract

In this paper we focus on the stochastic kinetic extension of the well-known Hodgkin-Huxley model of a biological neuron. We show the gradient descent algorithm for training of the neuron model. In comparison with training of the Hodgkin-Huxley model we use only three weights instead of nine. We show that the trained stochastic kinetic model gives equally good results as the trained Hodgkin-Huxley model, while we gain on more concise mathematical description of the training procedure. The trained stochastic kinetic model of neuron is tested in solving the problem of approximation, where for the approximated function the membrane potential obtained using different models of a biological neuron was chosen. Additionally, we present a simple application, in which the trained models of neuron connected with the outputs of a recurrent neural network form a system, which is used to calculate the Euler angles of an object's position in space, based on linear and angular acceleration, direction and the magnitude of Earth's magnetic field.

Keywords Hodgkin-Huxley model · Markov kinetic formalism · Gradient descent · AHRS · Attitude and heading reference system

1 Introduction

Neuron models which precisely describe the processes that take place on the membrane (in particular the Hodgkin-Huxley model and its kinetic extensions) can be expressed in the form commonly used for artificial neural networks. The biological neuron model consists of a current input and a voltage output which is coupled with the input; there are also parameters that can be treated as weights [5–7] and sigmoid functions that determine the activation functions.

In the Hodgkin-Huxley model considered in [5–7] nine parameters are used as weights. In this paper we show that if the stochastic kinetic model is used, it is possible to obtain equally good results of training and gain on less complex mathematical description of the training procedure, due to reducing the number of weights to three and replacing most of differential equations with 'add-and-multiply' rules. The gradient descent algorithm is used in the training procedure.

Researchers use models of a biological neuron in a number of applications, very often in the field of automatics and robotics. Usually these models exhibit only the spiking nature of a neural cell, without taking into account all of the ionic processes that take place in the neuron. Such spiking neuron models are applied in:

- robot control (cf. [1], where a neural network built with Izhikevich neuron models was tested on a model of an arm of a humanoid robot iCub),
- a neural PID controller, where a neuron model is expected to adjust parameters of the controller or to approximate differential and integral equations [28],
- creation of a visual attention map [29],
- general path planning [13, 21, 30].

The neuron model commonly used in the above applications is the Izhikevich model, presented among others in [14]. This model is described with only two equations:

$$\begin{aligned}v' &= 0.04v^2 + 5v + 140 - u + I, \\u' &= a(bv - u),\end{aligned}\tag{1}$$

where v is the potential on the neuron's membrane, u is the membrane recovery variable and I is the input current (stimulus), while a and b represent the time scale and sensitivity of the recovery variable, respectively. In our paper, however,

✉ Aleksandra Świetlicka
aleksandra.swietlicka@put.poznan.pl

¹ Faculty of Computing, Institute of Automatic Control and Robotic, Poznan University of Technology, ul. Piotrowo 3A, 60-965 Poznań, Poland

we will consider the stochastic kinetic model of biological neuron, which besides the spiking nature of neuron, takes into account ionic processes that take place on the membrane.

In this paper we present the gradient descent algorithm for training the stochastic kinetic model of neuron. The trained model is used to control the process of calculating the Euler angles of an object in space (Roll, Pitch, Yaw) based on data collected from three sensors: accelerometer, gyroscope and magnetometer. These sensors are part of the Attitude and Reference Heading System (AHRS) and are created in Micro Electro-Mechanical Systems (MEMS) technology. The AHRS enables calculation of the Euler angles, but the AHRS algorithm is computationally expensive in view of matrix calculations, hence the idea of using an artificial neural network combined with a model of a biological neuron, which is less complex than commonly used algorithms.

The same algorithm of gradient descent was already used for the training of a model of a dendritic structure of the biological neuron given with the following equation [12]:

$$\frac{a}{2R} \frac{\partial^2 V}{\partial x^2} = C \frac{\partial V}{\partial t} + g_{Na} m^3 h (V - V_{Na}) + g_K n^4 (V - V_K) + g_L (V - V_L) \quad (2)$$

This means adopting the algorithm of gradient descent in the structure that can be treated as a biological neural network (not just a single neuron). The way of discretization and implementation of this kind of dendritic structure of biological network is presented in [26]. The procedure of the training of the stochastic kinetic model of a biological neural network and application of the trained model in an image processing task are shown in [23].

The main objective of the research presented in this paper is focused on determining an object's position in space. Our research also includes self-testing of robot's sensors [15]. We are still expanding these calculations with new ideas. Models of biological neuron and biological neural networks are of our interest. We are trying to apply them to solve our problems and to analyse their capabilities, hence the presented research.

The paper is organized as follows: in Section 2 shows the Hodgkin-Huxley model and its extension to the deterministic and stochastic kinetic model. Section 3 presents the gradient descent algorithm for training of the stochastic kinetic model. Section 4 focuses on experimental results, while Section 5 shows applications of a trained model of the biological neuron. Finally, in Section 6, a short summary is provided.

2 Model of the Biological Neuron

In this section we will provide a short description of the Hodgkin-Huxley neuron model and its deterministic and stochastic kinetic extensions.

2.1 The Hodgkin-Huxley Model of the Biological Neuron

The neuron model created by Alan Lloyd Hodgkin and Andrew Huxley in 1952 is based on the idea that the potential on the membrane can be modeled with an equivalent circuit and described with the following equation [2, 9, 12, 16, 17]:

$$C \frac{dV}{dt} = I - g_{Na} m^3 h (V - V_{Na}) - g_K n^4 (V - V_K) - g_L (V - V_L), \quad (3)$$

where I is the total membrane current density (input of the neuron), V is the potential (output of the neuron), C is capacitance, while g_{Na} , g_K and g_L are conductivities and V_{Na} , V_K , V_L are the reverse potentials of specific types of ions - sodium, potassium and chloride, respectively. Values of the conductivities and reverse potentials are shown in Table 1.

The membrane is built from channels which consist of small gates, which in turn control the movement of ions between the interior and exterior of the neuron. In the Hodgkin-Huxley approach it is assumed that gates can be in either of two states, permissive or non-permissive. Dependence between these states is described with the following equations:

$$\begin{aligned} \frac{dn}{dt} &= \alpha_n(V) (1 - n) - \beta_n(V) n, \\ \frac{dm}{dt} &= \alpha_m(V) (1 - m) - \beta_m(V) m, \\ \frac{dh}{dt} &= \alpha_h(V) (1 - h) - \beta_h(V) h, \end{aligned} \quad (4)$$

where the first term ($\alpha_i(V) (1 - i)$) represents the transition of a particular gate from the non-permissive to permissive state, while the second term ($\beta_i(V) i$) – the transition from the permissive to non-permissive state.

Sigmoid functions $\alpha(V)$ and $\beta(V)$ are of the following form:

$$\begin{aligned} \alpha_n(V) &= \frac{0.01 \cdot (10 - V)}{\exp\left(\frac{10 - V}{10}\right) - 1} & \beta_n(V) &= 0.125 \cdot \exp\left(-\frac{V}{80}\right) \\ \alpha_m(V) &= \frac{0.1 \cdot (25 - V)}{\exp\left(\frac{25 - V}{10}\right) - 1} & \beta_m(V) &= 4 \cdot \exp\left(-\frac{V}{18}\right) \\ \alpha_h(V) &= 0.07 \cdot \exp\left(-\frac{V}{20}\right) & \beta_h(V) &= \frac{1}{\exp\left(\frac{30 - V}{10}\right) + 1} \end{aligned} \quad (5)$$

Table 1 The parameters of Hodgkin-Huxley model

i	V_i [mV]	g_i [mS/cm ²]
Na	115	120
K	−12	36
L	10.6	0.3

2.2 Deterministic Kinetic Model of the Biological Neuron

A kinetic extension of the Hodgkin-Huxley model requires that processes which take place on the membrane are described with Markov kinetic schemes. The dynamics of the potential change on the membrane of the neuron can be written in the following form [22]:

$$C \frac{dV}{dt} = I - g_{Na} [m_3 h_0] (V - V_{Na}) - g_K [n_4] (V - V_K) - g_L (V - V_L) \quad (6)$$

In the kinetic model of neuron, sodium and potassium gates can be in eight and five states, respectively. The relationship between these states is presented with Markov kinetic schemes of the following forms [22]:

$$\begin{array}{ccccccc} m_0 h_0 & \xrightleftharpoons[\beta_m]{3\alpha_m} & m_1 h_0 & \xrightleftharpoons[\beta_m]{2\alpha_m} & m_2 h_0 & \xrightleftharpoons[\beta_m]{\alpha_m} & m_3 h_0 \\ \alpha_h \updownarrow \beta_h & & \alpha_h \updownarrow \beta_h & & \alpha_h \updownarrow \beta_h & & \alpha_h \updownarrow \beta_h \\ m_0 h_1 & \xrightleftharpoons[\beta_m]{3\alpha_m} & m_1 h_1 & \xrightleftharpoons[\beta_m]{2\alpha_m} & m_2 h_1 & \xrightleftharpoons[\beta_m]{\alpha_m} & m_3 h_1 \end{array} \quad (7)$$

$$\begin{array}{ccccccc} n_0 & \xrightleftharpoons[\beta_n]{4\alpha_n} & n_1 & \xrightleftharpoons[\beta_n]{3\alpha_n} & n_2 & \xrightleftharpoons[\beta_n]{2\alpha_n} & n_3 & \xrightleftharpoons[\beta_n]{\alpha_n} & n_4 \end{array} \quad (8)$$

The states of each of the kinetic schemes describe the permeability of the membrane with respect to particular types of ions. In the scheme for sodium gates, $[m_3 h_0]$ represents a fully permissive state in which the flow of sodium ions between the interior and the exterior of the neuron is possible. In the scheme for potassium gates, the permissive state is denoted by $[n_4]$ [27].

2.3 Stochastic Kinetic Model of the Biological Neuron

In the stochastic approach, the number of gates that in a given moment change the state is taken from the binomial distribution, which under certain assumptions can be approximated with the normal distribution [8]. Drawing the numbers of gates from a normal distribution is performed with the True Random Number Generator presented in [11].

Let us assume that at time t there are n_A gates in state A and n_B gates in state B . Coefficient r_{AB} represents the rate function that describes the transition of a single gate from state A to state B . Then we can determine the probability of transfer of one gate from state A to B between moments t and $t + \Delta t$ as $p = \Delta t \cdot r_{AB}$. Hence, for every step of time we can obtain the number of gates Δn_{AB} that transfer from state A to state B by choosing a random number from the binomial distribution, where $n = n_A$ and $k = \Delta n_{AB}$. If we assume that $\mu = \Delta n_{AB} \cdot p$ and $\sigma = \Delta n_{AB} \cdot p \cdot (1 - p)$ we can approximate the binomial distribution with the normal distribution [8].

A detailed description of the kinetic formalism can be found in [4].

3 Training Algorithm

It has already been shown [5–7] that it is possible to train the Hodgkin-Huxley neuron model by manipulating its parameters. In [5–7] $\alpha(V)$ and $\beta(V)$ are rewritten in the following form:

$$\begin{aligned} \alpha_n(V) &= \frac{0.01 \cdot (V_{\alpha_n} - V)}{\exp\left(\frac{V_{\alpha_n} - V}{10}\right) - 1} & \beta_n(V) &= 0.125 \cdot \exp\left(\frac{V_{\beta_n} - V}{80}\right) \\ \alpha_m(V) &= \frac{0.1 \cdot (V_{\alpha_m} - V)}{\exp\left(\frac{V_{\alpha_m} - V}{10}\right) - 1} & \beta_m(V) &= 4 \cdot \exp\left(\frac{V_{\beta_m} - V}{18}\right) \\ \alpha_h(V) &= 0.07 \cdot \exp\left(\frac{V_{\alpha_h} - V}{20}\right) & \beta_h(V) &= \frac{1}{\exp\left(\frac{V_{\beta_h} - V}{10}\right) + 1} \end{aligned} \quad (9)$$

where V_{α_n} , V_{α_m} , V_{α_h} , V_{β_n} , V_{β_m} and V_{β_h} together with g_{Na} , g_K and g_L constitute the weights of the model.

In our approach we use the gradient descent algorithm to train the stochastic kinetic model of the biological neuron [25]. This training method allows us to adjust the weights (in our case only g_{Na} , g_K , g_L), so that the potential waveform could fit the given target potential.

In the process of determining the value of the potential V according to the stochastic kinetic model, functions $\alpha(V)$ and $\beta(V)$ play an auxiliary role, while they take part in drawing numbers from a normal distribution. That is why we do not need to know the exact influence of parameters V_{α_n} , V_{α_m} , V_{α_h} , V_{β_n} , V_{β_m} and V_{β_h} on the functions $\alpha(V)$ and $\beta(V)$ and thus we do not treat them as weights, so their values would remain as default:

$$\begin{aligned} V_{\alpha_n} &= 10 & V_{\beta_n} &= 0 \\ V_{\alpha_m} &= 25 & V_{\beta_m} &= 0 \\ V_{\alpha_h} &= 0 & V_{\beta_h} &= 30 \end{aligned} \quad (10)$$

The gradient descent algorithm works as follows:

1. Define the problem (minimization of error function E).
2. Determine initial values of weights $w_1, w_2, \dots, w_n$ (in our case g_{Na}, g_K, g_L).
3. Determine the value of gradient of the error function:

$$\nabla E = \left[\frac{\partial E}{\partial w_1}, \frac{\partial E}{\partial w_2}, \dots, \frac{\partial E}{\partial w_n} \right]. \quad (11)$$

4. Update the values of weights according to the rule:

$$w_i \rightarrow w_i - \eta \nabla E(w_i), \quad (12)$$

(where η is the learning rate) until either the end condition is met:

$$e = |V - V^*| < \epsilon \quad (13)$$

or the imposed number of iterations is reached.

We can write the minimized function as the mean-square error [5–7]:

$$E = \frac{1}{T} \int_0^T \frac{1}{2} (V(t) - V^*(t))^2 dt, \quad (14)$$

where V is the potential obtained from the neuron model and V^* is the target of the training (target potential). In comparison with [5–7], where nine weights were assumed, we consider only three weights, namely g_{Na} , g_K and g_L rewritten in the form [5–7]:

$$\begin{aligned} w_1 &= g_{Na} = \bar{g}_{Na} e^{\tilde{g}_{Na}}, \\ w_2 &= g_K = \bar{g}_K e^{\tilde{g}_K}, \\ w_3 &= g_L = \bar{g}_L e^{\tilde{g}_L}, \end{aligned} \quad (15)$$

where $\bar{w}$ represents the default value of the considered parameter, while $\tilde{w}$ – the adopted weight and:

$$\begin{aligned} \bar{g}_{Na} &= 120, \\ \bar{g}_K &= 36, \\ \bar{g}_L &= 0.3. \end{aligned} \quad (16)$$

We can rewrite the equation for the dynamics of the potential change on the membrane in the following form:

$$\dot{V} = F(V, g_{Na}, g_K, g_L), \quad (17)$$

where

$$F = \frac{1}{C} (I - g_{Na} [m_3 h_0] (V - V_{Na}) - g_K [n_4] (V - V_K) - g_L (V - V_L)). \quad (18)$$

If we want to know the influence of particular parameters on Eq. 17, we can formulate a differential equation for each parameter:

$$\dot{Y}_i = \frac{\partial F}{\partial V} \cdot Y_i + \frac{\partial F}{\partial w_i}, \quad (19)$$

where

$$Y_i = \frac{\partial V}{\partial w_i}, \quad (20)$$

hence the differential equations system:

$$\begin{cases} \dot{Y}_1 = G \cdot Y_1 + \frac{1}{C} [m_3 h_0] (V - V_{Na}) \\ \dot{Y}_2 = G \cdot Y_2 + \frac{1}{C} [n_4] (V - V_K) \\ \dot{Y}_3 = G \cdot Y_3 + \frac{1}{C} (V - V_L) \end{cases} \quad (21)$$

where

$$G = \frac{\partial F}{\partial V} = -\frac{1}{C} (g_{Na} [m_3 h_0] + g_K [n_4] + g_L). \quad (22)$$

The gradient of the minimized function can be written in the form:

$$\frac{\partial E}{\partial w_i} = \frac{1}{T} \int_0^T (V(t) - V^*(t)) \frac{\partial V(t)}{\partial w_i} dt. \quad (23)$$

Update of the weights can be done either by updating the parameters after simulating the neuron model for time T and integrating the error gradient according to (23), or by the following scheme [5–7]:

$$T_a \Delta \tilde{w}_i = -\Delta \tilde{w}_i + \frac{1}{T} (V(t) - V^*(t)) \frac{\partial V(t)}{\partial w_i} \cdot \frac{\partial w_i}{\partial \tilde{w}_i}, \quad (24)$$

where

$$\tilde{w}_i = -\eta \tilde{w}_i \quad (25)$$

and $T_a = T$, $\eta = 0.01$. In the above procedure parameters are updated on-line with the running average of the gradient [5–7].

Fig. 1 Waveforms of potential: **a.** the target potential, **b.** the result of the trained Hodgkin-Huxley model, **c.** – the result of the trained stochastic kinetic model

4 Experimental Results

The first application of the trained stochastic kinetic model of neuron is the problem of approximation. As the target of the training (approximated function) we use the potential obtained from different models of a biological neuron (the Hodgkin-Huxley model of a neuron with a point-like structure and the Hodgkin-Huxley model of a neuron with a dendritic structure).

Figure 1 shows three potential waveforms. The first one is the potential obtained from the Hodgkin-Huxley model of neuron for $I = 40[\mu A]$ and it is set as the target of the training. The second and third waveforms were obtained in the training of the Hodgkin-Huxley and the stochastic kinetic model of neuron, respectively.

The results from the Hodgkin-Huxley and the stochastic kinetic model of neuron are similar. This means that we can replace the Hodgkin-Huxley model with the stochastic kinetic model. This will give us the implementation benefits related with the reduction of calculations resulting from the formulation of the model and from the reduction of the number of weights in the training process.

Fig. 2 The output of the stochastic kinetic model of neuron: **a.** the target potential, and the results for different values of ϵ : **b.** 10^{-4} , **c.** 10^{-6} , **d.** 10^{-7}

Additionally, Fig. 2 shows the behavior of a trained stochastic kinetic model for different values of ϵ . The target potential was obtained from the Hodgkin-Huxley model of a dendritic structure of the neuron [12]. The first waveform is the target potential, while the remaining ones are the waveforms of potential obtained during the training process. Note that if the value of ϵ is too small, then it distorts the results. It happens when the target potential is of the same order as ϵ . That is why it is important to properly choose the initial values of all parameters.

Figure 3 presents the results of training of the stochastic kinetic model, where the number of iterations was set to different values. With the increase in the number of iterations the accuracy of the trained neuron model improves. However, increasing the number of iterations above 200 does not give better results but only lengthens the time of training.

To emphasize the difference between the Hodgkin-Huxley and stochastic kinetic model of neuron, in Table 2 we present the times of training of these models for different numbers of iterations (hardware specification: 64-bit

Fig. 3 The output of the stochastic kinetic model of neuron for different numbers of iterations: **a.** the target potential, **b.** no training, **c.** 10 iterations, **d.** 50 iterations, **e.** 100 iterations, **f.** 200 iterations

Table 2 Comparison of times of training of the stochastic kinetic model and Hodgkin-Huxley model - times are given in seconds

No. of iterations	Time of training of the:	
	Hodgkin-Huxley model	stochastic kinetic model
no training	2.52	1.45
10 iterations	19.41	1.88
50 iterations	95.65	3.78
100 iterations	188.41	6.12
200 iterations	374.91	10.64
500 iterations	938.95	24.85

Windows operating system, 32 GB RAM, processor i7, 2.8 GHz). Line 'no training' correspond to the initialization time, which includes assignment of initial values to all of the parameters and calculating all the functions used for training. It is possible to observe that times of training of the stochastic kinetic model are much shorter than of the Hodgkin-Huxley model. These results conclusively show the advantage of training the stochastic kinetic model over training the Hodgkin-Huxley one.

Additionally, in our previous research we examined thoroughly the differences between a deterministic and stochastic kinetic model of a biological neuron. Among other indicators we have made a comparison of the number of gates that change their state in a specific time step for both versions of the model. In paper [24] we present our results which show that stochastic kinetic model due to the simplification of the differential equations to the mathematical rules can significantly reduce the training time without loss of the neuron's properties.

The generalization ability of the stochastic kinetic model of neuron has been thoroughly tested and confirmed in [27].

5 Application

The AHRS (Attitude and Heading Reference System) makes it possible to calculate the position of an object

in space. The system consists of sensors that provides information about the three degrees of freedom related with circular motion along axes x , y and z . It consists of the accelerometer, gyroscope and magnetometer – sensors created in the MEMS (Micro Electro-Mechanical Systems) technology [3]. The system makes it possible to determine the position of an object in space based on linear and angular acceleration, direction and the magnitude of Earth's magnetic field.

As we said earlier, determining the Euler angles using AHRS is computationally expensive. However, to control a real object it is necessary to perform all the calculations in real time. That is why we were seeking for other solutions, which would speed up the calculations. Hence the idea of using a recurrent neural network and since the results were not precise enough [18], we combined it with a trained model of the biological neuron.

We used the structure shown in Fig. 4, built from the Elman recurrent neural network and three models of a biological neuron. We tested many different structures of the Elman recurrent neural network and chose three for further tests, as they gave the best results.

The input of the neural network is a set of nine signals collected from three sensors (accelerometer, gyroscope and magnetometer) with respect to three axes. The output renders the values of three Euler angles (Roll, Pitch, Yaw). For the hidden layer we chose three options:

Fig. 4 The network used to obtain values of the Euler angles

Fig. 5 Data obtained from the accelerometer along x , y and z axes, respectively

Fig. 7 Data obtained from the magnetometer along x , y and z axes, respectively

Fig. 6 Data obtained from the gyroscope along x , y and z axes, respectively

Fig. 8 Euler angle along the x axis (Roll) obtained with: **a.** AHRS, **b.** Elman neural network, **c.** stochastic kinetic model of a biological neuron; **d.** The difference between **a** and **c**

Fig. 9 Euler angle along the y axis (Pitch) obtained with: **a.** AHRS, **b.** Elman neural network, **c.** stochastic kinetic model of a biological neuron; **d.** The difference between a and c

- 20 neurons in the hidden layer, 10 delays from the hidden layer to the input layer,
- 30 neurons in the hidden layer, 10 delays from the hidden layer to the input layer,
- 50 neurons in the hidden layer, 10 delays from the hidden layer to the input layer.

A detailed description of the Elman neural network can be found in [18, 19].

The outputs of the Elman network are connected with three models of a biological neuron. The task of trained models of a biological neuron is to smooth the waveform obtained from the Elman network.

Input data is shown in Figs. 5, 6, 7.

Figures 8, 9 and 10 show the values of the original Euler angles (expected values obtained with the AHRS algorithm) and values obtained with the Elman neural network and the trained stochastic kinetic model of neuron. Additionally in each figure we present the difference between the Euler

Fig. 10 Euler angle along the z axis (Yaw) obtained with: **a.** AHRS, **b.** Elman neural network, **c.** stochastic kinetic model of a biological neuron; **d.** The difference between a and c

angle and the output of the trained stochastic kinetic model of a biological neuron. This difference is calculated according to the equation:

$$e = |V^* - V|$$

where V^* is the target potential (expected value of the particular Euler angle) and V is the output of the trained stochastic kinetic model of neuron. Note that regardless of the angle type, this difference is always of the order 10^{-4} .

In Fig. 8 we additionally present the zoom of the chosen segments of waveforms presented in A and B, and A and C, to show that the output of the trained stochastic kinetic model of neuron is very precise.

Tables 3, 4 and 5 present root-mean-square errors (RMSE) between expected values of Euler angles and values of angles obtained using the Elman recurrent neural network, and Hodgkin-Huxley and stochastic kinetic models of biological neuron.

Training of the Elman recurrent neural network, depending on the number of neurons in the hidden layer:

Table 3 Root-mean-square error (RMSE) between expected values of the Euler angles and the values obtained using the Elman neural network (with 20 neurons in the hidden layer and 10 delays), Hodgkin-Huxley model and stochastic kinetic model

		Elman 20	HH	stochastic
TEST	ROLL	3.9072	2.0835e-06	2.3237e-06
	PITCH	3.2599	2.8292e-06	2.3233e-06
	YAW	3.7419	2.1126e-06	2.3230e-06
TEST	ROLL	4.0434	2.1700e-06	2.3226e-06
	PITCH	3.3308	2.8964e-06	2.3243e-06
	YAW	4.4965	2.2138e-06	2.3237e-06
X OSC	ROLL	2.3083	2.0464e-06	2.3236e-06
	PITCH	1.7467	2.8367e-06	2.3231e-06
	YAW	2.8281	2.0498e-06	2.3243e-06
Y OSC	ROLL	1.9970	2.0954e-06	2.3229e-06
	PITCH	0.8983	2.8678e-06	2.3233e-06
	YAW	2.3081	2.1185e-06	2.3239e-06
Z OSC	ROLL	2.7703	2.0804e-06	2.3241e-06
	PITCH	0.8844	2.8537e-06	2.3240e-06
	YAW	4.4231	2.0386e-06	2.3238e-06

- around 20 minutes for 20 neurons in hidden layer,
- over 6 hours for 30 neurons in hidden layer,
- over 4.5 days for 50 neurons in hidden layer.

Computation of one set of the Euler angles with Elman neural network takes approximately 0.03 s, independently of the number of neurons in the hidden layer. In case of the Hodgkin-Huxley model time of training was around 75 minutes, while of the stochastic kinetic model – around 20 minutes. Time of the computation of one set of the Euler

angles with both Hodgkin-Huxley and stochastic kinetic models was always 0.000199 s.

In case of the Elman recurrent neural network, the RMSE is lower for the structure of network with 30 neurons in the hidden layer than for 20 neurons. However, increasing the number of neurons in the hidden layer to 50 does not significantly decrease the value of the RMSE. Training the neural network with a bigger structure requires greater computational effort and is much more time consuming.

Table 4 Root-mean-square error (RMSE) between expected values of the Euler angles and the values obtained using the Elman neural network (with 30 neurons in the hidden layer and 10 delays), Hodgkin-Huxley model and stochastic kinetic model

		Elman 30	HH	stochastic
TEST	ROLL	1.8808	2.1426e-06	2.3243e-06
	PITCH	0.7289	2.8808e-06	2.3243e-06
	YAW	2.5508	2.1636e-06	2.3226e-06
TEST	ROLL	2.4956	2.1371e-06	2.3243e-06
	PITCH	0.6460	2.8474e-06	2.3237e-06
	YAW	2.2600	2.1884e-06	2.3226e-06
X OSC	ROLL	2.0321	2.0172e-06	2.3237e-06
	PITCH	0.7242	2.7854e-06	2.3243e-06
	YAW	3.8999	2.0520e-06	2.3232e-06
Y OSC	ROLL	2.5243	2.1454e-06	2.3241e-06
	PITCH	1.3084	2.8912e-06	2.3243e-06
	YAW	5.1249	2.1965e-06	2.3243e-06
Z OSC	ROLL	3.8929	2.0592e-06	2.3241e-06
	PITCH	0.8638	2.8438e-06	2.3240e-06
	YAW	6.2121	2.1130e-06	2.3238e-06

Table 5 Root-mean-square error (RMSE) between expected values of the Euler angles and the values obtained using the Elman neural network (with 50 neurons in the hidden layer and 10 delays), Hodgkin-Huxley model and stochastic kinetic model

		Elman 50	HH	stochastic
TEST	ROLL	7.9550	2.1251e-06	2.3239e-06
		0.9635	2.8727e-06	2.3239e-06
		17.8374	2.1988e-06	2.3243e-06
	PITCH	8.0728	2.2086e-06	2.3211e-06
		0.8428	2.9311e-06	2.3243e-06
		18.6244	2.2636e-06	2.3199e-06
	YAW	7.8666	2.2136e-06	2.3229e-06
		0.6345	2.9184e-06	2.3237e-06
		15.7388	2.2550e-06	2.3232e-06
X OSC	ROLL	4.2710	2.1595e-06	2.3234e-06
	PITCH	1.0300	2.8790e-06	2.3242e-06
	YAW	14.6253	2.2089e-06	2.3241e-06
Y OSC	ROLL	5.1965	2.1298e-06	2.3241e-06
	PITCH	0.7624	2.8712e-06	2.3240e-06
	YAW	15.3884	2.1722e-06	2.3238e-06
Z OSC	ROLL			
	PITCH			
	YAW			

The structure with 30 neurons in the hidden layer is satisfactory, especially since the results obtained using the Elman neural network are further processed.

Although the results obtained using the trained Hodgkin-Huxley model are similar to those obtained using the trained stochastic kinetic model, it is important to remember that in the case of the stochastic kinetic model there is the aspect of modelling the unpredictable nature of a biological neuron. Additionally, training of the stochastic kinetic model is much faster than of the Hodgkin-Huxley one (Table 2).

6 Discussion

The algorithm presented in this paper simplifies and speeds up the process of training of the stochastic kinetic model of neuron. The simulation of the model is faster; moreover, the reduction in the number of weights results in faster training process [10, 24].

The presented application confirms the usefulness of introduction of the training process into biological neural networks. These networks exhibit associative properties [20], which gives them a great advantage over conventional artificial neural networks. A trained model of the biological neuron solves the problem of finding the Euler angles based on samples taken from an accelerometer, magnetometer and gyroscope. An artificial neural network used in the same problem did not return satisfactory results.

This paper concerns mainly the procedure of the training of the stochastic kinetic model and not the AHRS algorithm itself. We think that it is necessary to continue this research, where presented in this paper solutions would be compared with other AHRS algorithms.

It is also necessary to consider other architectures of the artificial neural networks, while in this paper we focus only on shallow structures of the Elman recurrent neural networks. Presented in this paper Elman neural network is also considered in our other research projects that concern object's position in space. At first we used these neural networks to obtain Euler angles in [18] and then combined them with a model of a biological neuron. In our other research [15], we examined also other structures of neural networks in order to detect sensor failure in a robot. One of the obvious future steps is to consider other machine learning methods, especially deep neural networks and among them graph neural networks.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

1. Bouganis, A., Shanahan, M.: Training a spiking neural network to control a 4-dof robotic arm based on spike timing-dependent plasticity. In: Proceedings of the 2010 International Joint Conference on Neural Networks (IJCNN), pp. 1–8 (2010)
2. Bower, J.M., Beeman, D.: The Book of GENESIS, Exploring Realistic Neural Models with the GENeral NEural SIMulation System. Internet Edition (2003)
3. Cai, G., Chen, B.M., Lee, T.H.: Unmanned Rotocraft Systems. Advances in Industrial Control. Springer, Berlin (2011)
4. Destexhe, A., Mainen, Z.F., Sejnowski, T.J.: Synthesis of models for excitable membranes, synaptic transmission and

- neuromodulation using a common kinetic formalism. *J. Comput. Neurosci.* **3**(1), 195–230 (1994)
5. Doi, S., Onoda, Y., Kumagai, S.: Parameter estimation of various Hodgkin-Huxley-type neuronal models using a gradient-descent learning method. In: *SICE 2002, Proceedings of the 41st SICE Annual Conference*, vol. 3, pp. 1685–1688 (2002)
 6. Doya, K., Selverston, A.I.: A learning algorithm for Hodgkin-Huxley type neuron models. In: *Proceedings of 1993 International Joint Conference on Neural Networks*, 1993. *IJCNN '9293-Nagoya*, vol. 2, pp. 1108–1111 (1993)
 7. Doya, K., Selverston, A.I., Rowat, P.F.: A Hodgkin-Huxley type neuron model that learns slow non-spike oscillations. In: *NIPS'9293*, pp. 566–573 (1993)
 8. Feller, W.: *An Introduction to Probability Theory and its Applications*, volume 1, chapter VII. Wiley, New York (1968)
 9. Gerstner, W., Kistler, W.: *Spiking Neuron Models. Single Neurons. Populations, Plasticity*, Cambridge (2000)
 10. Gugała, K., Figas, A., Jurkowlanec, A., Rybarczyk, A.: Parallel simulation of stochastic denritic neurons using nvidia gpu with cuda c. In: *2011 Proceedings of the 18th International Conference Mixed Design of Integrated Circuits and Systems (MIXDES)* (2011)
 11. Gugała, K., Świetlicka, A., Burdajewicz, M., Rybarczyk, A.: Random number generation system improving simulations of stochastic models of neural cells. *Computing* **95**(1), 259–275 (2013)
 12. Hodgkin, A.L., Huxley, A.F.: A quantitative description of membrane current and its application to conduction and excitation in nerve. *J. Phys.* **117**(4), 500–544 (1952)
 13. Hopfield, J.J.: Neurodynamics of mental exploration. *Proc. Natl. Acad. Sci. USA* **107**(4), 1648–1653 (2010)
 14. Izhikevich, E.M.: Simple model of spiking neurons. *IEEE Trans. Neural Netw.* **14**(6), 1569–1572 (2003)
 15. Kapela, R., Świetlicka, A., Kolanowski, K., Pochmara, J., Rybarczyk, A.: A set of dynamic artificial neural networks for robot sensor failure detection. In: *11th International Workshop on Robot Motion and Control 2017 (RoMoCo)*, pp. 199–204 (2017)
 16. Keener, J., Sneyd, J.: *Mathematical Physiology*, volume 8 of *Mathematical Biology*. Springer, Berlin (1998)
 17. Koch, C.: *Biophysics of Computation: Information Processing in Single Neurons*, chapter, VI. Oxford University Press, New York (1999)
 18. Kolanowski, K., Świetlicka, A., Kapela, R., Pochmara, J., Rybarczyk, A.: Multisensor data fusion using elman neural networks. *Appl. Math. Comput.* **319**, 236–244 (2018)
 19. Kolanowski, K., Świetlicka, A., Majchrzycki, M., Gugała, K., Karoń, I., Rybarczyk, A.: Nine-axis imu sensor fusion using the ahrs algorithm and neural networks. In: *ISTET International Symposium Theoretical Electrical Engineering*, Pilsen, Czech Republic, pp III–23–III–24, 24th–26th June 2013 (2013)
 20. Maass, W.: Networks of spiking neurons: the third generation of neural networks models. *Neural Netw.* **10**, 1659–1671 (1997)
 21. Saffari, A.: Unknown environment representation for mobile robot using spiking neural networks. *Trans. Eng. Comput. Technol.* **6**, 49–52 (2005)
 22. Schneidman, E., Freedman, B., Segev, I.: Ion channel stochasticity may be critical in determining the reliability and precision of spike timing. *Neural Comput.* **10**, 1679–1703 (1998)
 23. Świetlicka, A.: Trained stochastic model of biological neural network used in image processing task. *Appl. Math. Comput.* **267**, 716–726 (2015)
 24. Świetlicka, A., Gugała, K., Jurkowlanec, A., Śniatała, P., Rybarczyk, A.: The stochastic, markovian, Hodgkin-Huxley type of mathematical model of the neuron. *Neural Netw. World* **25**(1), 219–239 (2015)
 25. Świetlicka, A., Gugała, K., Karoń, I., Kolanowski, K., Majchrzycki, M., Rybarczyk, A.: Gradient method of learning for stochastic kinetic model of neuron. In: *ISTET International Symposium Theoretical Electrical Engineering*, Pilsen, Czech Republic, pages III–17–III–18, 24th – 26th June 2013 (2013)
 26. Świetlicka, A., Gugała, K., Pedrycz, W., Rybarczyk, A.: Development of the deterministic and stochastic markovian model of a dendritic neuron. *Biocybernetics Biomed. Eng.* **37**, 201–216 (2017)
 27. Świetlicka, A., Kolanowski, K., Kapela, R., Galicki, M., Rybarczyk, A.: Investigation of generalization ability of a trained stochastic kinetic model of neuron. *Appl. Math. Comput.* **319**, 115–124 (2018)
 28. Webb, A., Davies, S., Lester, D.: Spiking neural PID controllers. *Neural Inf. Process Lect. Notes Comput. Sci* **7064**, 259–267 (2011)
 29. Wu, Q.X., McGinnity, T.M., Maguire, L., Cai, R., Chen, M.: A visual attention model based on hierarchical spiking neural networks. *Neurocomputing* **16**, 3–12 (2013)
 30. Yanduo, Z., Kun, W.: Application of liquid state machines in robot path planning. *J. Comput.* **4**(11), 1182–1186 (2009)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Aleksandra Świetlicka graduated from Poznan University of Technology in Poland with Engineer's degree in Automatics and Management in 2008, Master of Science degree in Mathematics (specialization: Mathematic Methods in Computer Science) in 2008 and Master of Science degree in Automatics and Management (specialization: Reprogrammable Control Systems) in 2009. In January 2014 she received her PhD degree on the basis of defended in December 2013 dissertation entitled "Stochastic model of biological neural network with use of Markov kinetic schemes". Currently she is employed as an assistant professor at the Poznan University of Technology.

Her area of research interests covers primarily models of artificial and biological neural networks. She creates mathematical description of biological neuron and neural networks and analyzes their properties in relation to conventional artificial neural networks. In her research she consider both shallow and deep structures of artificial neural networks and she expands her area of interests with new machine learning methods. She also uses artificial neural networks to solve numerous problems in image and signal processing. Her latest projects concerned real-time detection of events that occur in sports broadcasts, estimation of angular speed of synchronous generator based on stator voltage, designation of Euler angles in the Attitude and Heading Reference System for control of unmanned aerial vehicle (quadcopter) and proteogenomic analyses of cancer. Her latest project concern knowledge base of real estate, which is guided by the Wrocław Institute of Spatial Information and Artificial Intelligence in Poland.

Krzysztof Kolanowski graduated from Poznan University of Technology in Poland with Master of Science degree in Automatics and Management in 2010. At this moment he is a research assistant at the Institute of Automatic Control and Robotics at the same University.

His current line of research is the use of artificial intelligence for multi-sensor data fusion and self-testing for detecting irregularities in multi-sensor data collection systems. An additional direction of his research is the hardware implementation of artificial neural networks on embedded systems.

He has published several papers about these subject both in electronic and printed formats.

Rafał Kapela began his research career during his Master's studies at Poznan University of Technology (PUT). Initially he was involved in projects connected to control and robotics, as well as artificial intelligence. During that time he proposed new obstacle avoidance algorithms for anthropomorphic robot manipulators, and a gesture recognition system based on neural networks. After receiving the title of Master of Science in year 2005, he was employed in the Department of Computer Engineering at PUT as a PhD fellow. That was the moment when he started professional research on the topic which is his passion—digital image and video processing. During the few years he held this assistant position, he was involved in a range of scientific projects in many areas, including hardware image/signal processing, hardware implementations of neural networks, and an analysis of RNA chains for medical purposes. His main focus, however, was on image content description based on visual descriptors and artificial intelligence. After receiving a PhD degree in August 2009 he had the opportunity to join a new team of design-for-test engineers at Mentor Graphics, located in Poznan. After this Rafał joined CLARITY – a research centre in Dublin City University as a Marie Curie research fellow with his individual research project for real-time sports event detection system called VISION. After returning to Poland he was involved in number of scientific and industrial research projects including: AXIOM (an open-hardware 4K camera), deep learning based automatic asphalt surface categorization, FPGA-based stereovision system and many others featuring FPGAs and GPUs as the universal hardware accelerators for real-time embedded vision processing tasks.